Screencast: 12-package-management.webm or 12-package-management.mp4

LaUSAH REFERENCE - Chapter 12, Software Installation and Management, Section 12.5, Managing Packages
TLCL REFERENCE - Chapter 14, Package Management

What you may already know

- How do you install software on Microsoft Windows?
- How do you tell what software is installed on Microsoft Windows?
- How about on Mac OS X?

Using rpm - lower level tool written in C

- Query packages:
  - -q, -qi (info), -qd (docs), -qc (configs)
  - -qa : query all
  - -qf - query file, what package a file came from
  - -ql - query list, list files from a package
- Verify packages: -V, no output is good
  - S file Size differs
  - M Mode differs (includes permissions and file type)
  - 5 MD5 sum differs
  - D Device major/minor number mismatch
  - L readLink(2) path mismatch
  - U User ownership differs
  - G Group ownership differs
  - T mTime differs
- Install software: -ivh (verbose with hashmarks)
- Upgrade: -Uvh, -Fvh
- Remove: -e
- Arch: rpm -qa --qf "%{n}.%{arch}\n"
- Special flags: --force, --nodeps, --justdb, --rebuilddb

Using dnf - higher level tool written in python

- install, localinstall
- update, localupdate, check-update
- remove or erase
- groups : grouplist, groupinstall
- list : installed, updates
- info {package-name}
- clean : all, packages, metadata
- provides or whatprovides
- search
- repolist
- Files:
  - /etc/dnf/dnf.conf : exclude=whatever*
  - /etc/yum.repos.d/*.repo : enabled=0/1, gpgcheck=0/1
  - /var/cache/dnf/

Where to find additional software

Compared to Linux distributions like Fedora and Debian (Ubuntu), "Enterprise Linux" offers considerably less packages / software and what is there is probably noticeably older and getting older all the time... with the 7 to 10 year life span. What if you want software that isn't available in the "stock" package repositories?

Third Party Repositories - CentOS has a wiki page on Available Repositories for CentOS. **EPEL** (Extra Pckages for Enterprise Linux) is done by the Fedora folks. Other repositories exist mainly to provide fresher web stacks (Python, Perl, PHP, MySQL / MariaDB, etc) including the Red Hat's **SCL** (Software CoLlections repository that CentOS rebuilds (side-by-side installs under /opt). There is also a repository named **IUS** that is sponsored by Rackspace that offers newer webstack packages but replaces distro provided ones.

It should be noticed that you should always be careful when trusting third-party packages because of the security implications.

Lastly, it is always possible to install software from source. Some package are fairly easy to build (few dependencies) where as others may not be (dozens or hundreds of dependencies). Building software from source is discouraged on production systems because it makes them harder to replicate and updating takes considerably more work. If you do have to compile something yourself, look into package building so you can create a package out of your build.

Other package managers

rpm came from Red Hat, and since the other two "original" distros are Slackware and Debian it only stands to reason that they would probably have their own package management systems.

Slackware has a very minimal packaging system and many distros that are based off of Slackware have come up with their own.

Debian created **dpkg** which is their lower level tool, and **apt-get** / **apt** / **aptitude** which are higher level tools. Their packages end with the .deb extension.

rpm / **dpkg** and **dnf / apt-get** certainly are not the end-all-be-all systems. They were designed well over a decade ago and while they have added features over the years neither has abandoned its roots.

Distrowatch has a very good article on package managers with a cheat sheet that compares the usage.

Graphical package managers

Like with most complex command line tools, there are graphical front-ends available for most package management systems. The most popular one for .deb based systems is Synaptic. For .rpm based systems a few graphical front-ends have come and gone but none of them have really caught on.

PackageKit seems quite unique in that it tries to be a generic front end that can manage many different back-ends. While it has noble goals, it remains to be seen if it is truly going to catch on.

Distro Agnostic Packaging?

There are several efforts to make distro agnostic software packaging. Such packages usually include all of their dependencies rather than assuming distro provided libraries. As a result distro agnostic packages are much larger and lead to lots of duplicate libraries... but they do offer a lot of convenience for software packagers. Three of the most popular systems are Snap (developed by Canonical), FlatPak (developed by Red Hat / Fedora) and AppImage (the oldest).

Additional Resources

Distrowatch's Package Management Cheatsheet file
http://distrowatch.com/dwres.php?resource=package-management